Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Nika Bric

Vizualizacija besedil ljudskih pesmi

diplomsko delo

univerzitetni študijski program prve stopnje
računalništvo in informatika

Mentor: doc. dr. Matija Marolt

Ljubljana 2015

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

V diplomski nalogi izdelajte vizualizacijo zbirke besedil ljudskih pesmi. Raziščite specifiko ljudske pesmi in izdelajte vizualizacijo, ki naj temelji na večnivojskem prikazu razmerij med variantnimi tipi in geografskem položaju zajema posameznih variant. Vizualizacija naj temelji na spletnih tehnologijah.

Izjava o avtorstvu diplomskega dela

Spodaj podpisana Nika Bric sem avtorica diplomskega dela z naslovom:

Vizualizacija besedil ljudskih pesmi

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelala samostojno pod mentorstvom doc. dr. Matije Marolta,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 11. septembra 2015 Podpis avtorja:

Zahvaljujem se mentorju doc. dr. Matiji Maroltu za svetovanje in pomoč. Alji Debeljak, Maju Smerkolu, Dušanu Kalanju in Jakobu Ihanu za oporo in občasno pomoč.

Kazalo

Seznam uporabljenih izrazov

izraz	razlaga
Apache	Apache HTTP strežnik
AngularJS	ogrodje za delo z JavaScriptom na principu MVC
CSV	Comma Separated Values, format za podatkovne tabele
D3.js	JavaScript knjižnica za delo s podatki (HTML, CSS, SVG)
DOM	Document Object Model, konvencija za interakcijo z X/HTML)
točkovni graf	graf, kjer se vsaka pojavitev ali dogodek označi s piko
gmaps.js	JavaScript knjižnica za delo z Google Maps
HTML	HyperText Markup Language, osnova spletnih strani
JavaScript	programski jezik za stran odjemalca
jQuery	JavaScript knjižnica
JSON	JavaScript Object Notation, format za prenos podatkov
LAMP	skupek orodij za delo s strežnikom
Linux	odprtokodni operacijski sistem
marker	značka, ki označuje lokacijo v Google Maps
MDS	Multi Dimensional Scaling, algoritem
PHP	Hypertext Preprocessor, programski jezik
Python	splošno namenski programski jezik
SVG	Scalable Vector Graphics, grafični format
Ubuntu	distribucija OS Linux
varianta	različica pesmi
variantni tip	skupina pesmi s približno enako vsebino
Windows	Microsoftov operacijski sistem
XML	EXtenstible Markup Language, podatkovni format

Povzetek

Diplomsko delo poskuša del zbirke slovenskih ljudskih pesmi prikazati s pomočjo interaktivne vizualizacije podatkov. Vizualizacija podatkov je predstavitev seta podatkov v slikovni in/ali grafični obliki, in uporabniku omogoči lažje delanje povezav in razbiranje pomena podatkov. Vizualizacija za glavno podatkovno zbirko uporabi seznam variantnih tipov ljudskih pesmi s pripadajočimi variantami.

Delo je potekalo tako, da je bil najprej izdelan načrt funkcionalnosti in izgleda aplikacije, ki je upošteval cilje. Sledila je obdelava podatkov, ki so bili potrebni za realizacijo izdelka, nato pa še sama implementacija. Orodja in tehnologije, v katerih je bila aplikacija napisana, so omenjene v diplomskem delu.

Opisana je tudi zgodovina ljudske pesmi s kratkim pregledom zbiranja besedil in melodij v Sloveniji. Ker se praktični del diplome nanaša na klasifikacijo ljudskih pesmi, so razložene nekatere osnovne strukture.

Rezultat diplome je delujoča aplikacija, ki bi potrebovala še nekaj iteracij, preden bi bila resnično uporabna, vendar pa lahko služi za dobro oporo nadaljnjemu razvoju.

Ključne besede: vizualizacija, vizualizacija podatkov, ljudske pesmi, besedila pesmi, interaktivno, d3.js.

Abstract

This thesis attempts to show a part of a collection of Slovenian folks songs using interactive data visualization. Data visualization is a presentation of a data set in a pictorial or graphic format, and it makes it easier for a user to grasp the meaning of data and the connections present. This visualization is using a list of folk song variant types with their respective variants as its data set.

The work started with designing a plan for the functionalities and the look of the application that suited the goals. The data needed for the realization of the program was processed and after that followed the implementation itself. Tools and technologies used to make the application are listed in the thesis.

The thesis includes a description of the history of folks songs and a short overview of collecting of lyrics and melodies in Slovenia. Some basic folk songs structures are explained, because the application is based on them.

The result is a working application, which needs a few more iterations before it can be truly useful, but it can serve well as a basis for further development.

Keywords: visualization, data visualization, folk songs, song lyrics, interactive, d3.js.

Poglavje 1

Uvod

V zadnjih nekaj letih se je močno povečalo število načinov merjenja podatkov, k čemur je pripomogla poenostavljena in dostopnejša uporaba orodij za merjenje. Ker vse več ljudi ustvarja oz. generira svoje podatke, se jih nabira vedno več. Termin, ki se uporablja za ogromno količino teh podatkov, je Big Data. [7] Od sporočil na socialnih omrežjih do naprav, ki merijo telesne funkcije, pa različnih logov, ki jih hrani programska oprema (naj bo to zakonsko določeno ali ne), so podatki najrazličnejših oblik in namembnosti. Večina teh podatkov ni strukturiranih (80%). [6] Da lahko iz njih dobimo informacije, ki nam nekaj povedo, jih je potrebno obdelati. Zato se v zadnjem času izjemno povečuje potreba po podatkovnih znanstvenikih (*data scientist*).

Veliko načinov za uporabo teh podatkov ne potrebuje vizualizacije, če pa jih želimo predstaviti v obliki, ki je preprosto razumljiva človeku (tudi nestrokovni javnosti), je vizualizacija ključnega pomena. Vizualizacija podatkov pomeni predstavitev podatkov v slikovni ali grafični obliki. [4] Na ta način ljudje lažje in hitreje zaznamo pomen podatkov, kot če bi jih brali v pisni ali tabelarični obliki. Bolj preprosta je tudi primerjava različnih podatkov, saj lahko z vpeljavo interaktivnosti spreminjamo parametre prikaza oz. na istem grafu ali sliki prikažemo več setov podatkov. Omogoča nam prikaz več dimenzij in oblik podatkov (npr. video, glasba). Pogosta tehnika je ani-

miranje grafa ali slike ob spreminjanju enega parametra podatkov, npr. časa. Tako dobimo boljši vpogled v spreminjanje določene spremenljivke skozi čas, po možnosti pa tudi v razloge za te spremembe in potencialno napoved za prihodnost.

V diplomskem delu bom poskusila združiti področji ljudskega slovstva in vizualizacije podatkov. Cilj dela je spletna aplikacija, ki vsebuje smiselno interaktivno vizualizacijo seznama ljudskih pesmi, ki vsebuje več razredov grupiranja besedil. V delu bom predstavila zgodovino ljudske pesmi, oris zbiranja pesmi v Sloveniji ter kratek opis nekaterih lastnosti, ki so značilne za ljudske pesmi. Nato bom opisala izdelavo vizualizacije, od orodij, ki sem jih uporabila, do podatkov, načrta in implementacije. Zaključila bom s krajšim komentarjem narejenega, opisom možnosti za izboljšave in nadaljnjega dela.

Poglavje 2

Ljudska pesem

2.1 Ozadje in zgodovina

Skozi ljudsko pesem in glasbo se je človeška rasa izražala že od razvoja jezika naprej. Pesem in glasba sta se prenašali ustno. Večji del prebivalstva pred 20. stoletjem ni bil pismen, zato so se tovrstnih kulturnih vsebin naučili na pamet. [5] Pesmi so se tako prenašale iz generacije v generacijo. Najstarejši del kulturne dediščine slovenskega naroda je prav ljudska pesem. [10]

Besedila so večinoma povezana z narodno ali lokalno kulturo; pogosto so govorila o zgodovinskih ali osebnih dogodkih. [5] Čeprav večina ljudi misli, da so ljudske pesmi vedno narečne, v resnici ni tako. Ker so se pesmi prenašale iz kraja v kraj, v njih ni ostalo toliko narečnih posebnosti. Za tiste pesmi, ki imajo bolj lokalen pomen, se je narečje bolj obdržalo, saj drugih pokrajin in krajev vsebina ni toliko zanimala, in se pesem ni razširila. Bolj narečno močne pesmi so značilne za obrobne pokrajine. Vsebinsko so nekatere pesmi značilne za določene pokrajine, npr. pesmi o naravi v Reziji, zdravice v vinorodnih okoliših, dolge pripovedne pesmi na Gorenjskem, da so si krajšali čas pri enoličnem skupnem delu. Velik del slovenske ljudske pesmi zadeva religiozno vsebino. Te pesmi pogosto spremljajo verske praznike, saj govorijo o dogodkih, ki jih praznik praznuje. Navkljub vsem razlikam se jasno vidi,

da ljudska pesem povezuje vse Slovence. [10]

Zaradi pomanjkanja zapisovanja in dolgega obdobja prenašanja za večino pesmi ustnega izročila avtor ni znan. Tako tovrstna umetnost tudi dobi pridevnik "ljudska", saj "lastniki"pesmi postanejo ljudje, ki jo pojejo, ponavadi lokalno ali narodno prebivalstvo. Nekatere definicije ljudske pesmi izključujejo tista dela, katerih avtor je znan. [11]

Značilnost ljudske pesmi, ki jo razlikuje od t.i. umetne poezije, je tudi to, da je neločljivo povezana z glasbo. Melodija prvič zelo pripomore k lažji zapomnljivosti besedila, kar je bilo v času, ko se jih ni zapisovalo, precej pomembno. Melodija se ujema z ritmom besedila, sčasoma pa je povzročila tudi, da so se spremenili ali na novo razvili verzni obrazci, kitice, ter nekatere ritmične posebnosti. Melodija pevcu, ki se zmoti v besedilu, tudi pomaga, da besedilo zimprovizira, in obdrži podobo pesmi. Ko se je začelo zbiranje oz. zapisovanje ljudskih pesmi, je pogosto prišlo do napak v ritmu pri narekovanju, če so pevci poskušali besedilo podati brez melodije.

2.2 Zbiranje

Prvo znano zbirko slovenskih ljudskih pesmi je zbral Dizma Zakotnik okrog leta 1775. Sestavljalo jo je pet besedil pripovednih pesmi. Zbirka se na žalost ni ohranila; vsebino delno poznamo zaradi Marka Pohlina, ki je bil tudi pobudnik zbiranja in je idejo dobil pri nemških in angleških zapisovalcih pesmi. Valentin Vodnik je zbral okoli 150 besedil, njegov prijatelj Jožef Rudež pa okoli 45. [23] Ko je nekaj zapisov Andreja Smoleta s Prešernovimi popravki izšlo v Kranjski Čbelici v 30-ih letih 19. stoletja, se je navdušilo veliko novih zbiralcev. Med njimi je bil najpomembnejši Stanko Vraz, ki se je zavedal pomembnosti melodije in jo je edini tudi zapisoval.

Okoli leta 1865 se je pojavila ideja, da bi izdali slovenske ljudske pesmi

iz cele države. Slovenska matica, ki naj bi zbirko izdala, je imela precejšnje težave z iskanjem urednika. Leta 1887 je Karel Štrekelj sestavil podroben načrt izdaje, v katerem je tudi prvič v slovenščini uporabil besedo "folklora". Zahteval je tudi, naj vsi zapisovalci zapišejo tudi melodijo. Prvi snopič je bil izdan leta 1895, zadnji, šestnajsti, pa leta 1923. [24]

V začetku zbiranja ljudskih pesmi seveda ni bilo na voljo snemalne opreme, zato so lahko besedilo oz. note samo zapisali. Pri teh je prihajalo do marsikaterih napak. Kot je bilo omenjeno zgoraj, ljudje niso pravilno narekovali besedila brez melodije. Tudi zapisovalci sami so se ponekod napačno lotili zapisovanja. Nekateri so želeli zapisati pesem natanko tako, kot jo je pevec zapel, z narečjem vred, ker pa na kraju zapisovanja samem takšna natančnost ni bila mogoča, so zapis kasneje popravljali. Takšne zapise je težko brati, kaj šele po njih peti. V drugo skrajnost so šli zapisovalci, ki so pesmi popravljali, tako da so bile bližje umetni pesmi, kar je prav tako narobe prikazalo dejansko stanje. [10] Primer: Prešeren je nekatera besedila v celoti prepesnil.

Prvi zvočni posnetki so nastali leta 1913, ko je ruska folkloristka J. E. Lineva na fonograf posnela okoli 100 pesmi, večinoma z Dolenjske in Gorenjske. [10]

Leta 1934 je bil ustanovljen Folklorni inštitut, ki je ponovno omogočil načrtno zbiranje pesmi, kar je bil po vojni prepuščeno posameznim ljubiteljem. Dvajset let kasneje je inštitut z večjim številom sodelavcev in magnetofonom zbiral pesmi po celotnem etničnem območju Slovenije. Zbiranje in objavljanje se je sčasoma razširilo tudi na radio. Najpomembnejša je bila oddaja z naslovom Slovenska zemlja v pesmi in besedi, ki obstaja še danes. [1] Sčasoma se je inštitut pridružil Slovenski akademiji znanosti in umetnosti. [2] Zdaj deluje pod imenom Glasbenonarodopisni inštitut in svoje delovanje posveča raziskovanju, zbiranju in hranjenju avtohtone slovenske ljudske umetnosti. Velik del zaslug za ogromno že opravljenega dela na tem področju pa

gre tudi individualnim zbirateljem in raziskovalcem.

2.3 Klasifikacija in struktura ljudske pesmi

2.3.1 Razdelitev

V slovenščini imamo sicer enak izraz - pesem - za besedila, ki se pojejo, in tista, ki se recitirajo, oz. imajo močen poudarek na ritmu, vendar pa jih vseeno ločimo. Pod ljudsko pesem lahko spadata obe vrsti.

Po vsebini ločimo ljudske pesmi na več načinov, npr. na lirske in epske oz. izpovedne in pripovedne. Pomembnejše vsebinske skupine v Sloveniji so ljubezenske pesmi, svatovske pesmi, mrliške pesmi, plesne pesmi, vojaške pesmi, pesmi o poklicih, zabavne pesmi, koledniške pesmi (pojejo se ob praznikih), nabožne in romarske pesmi, pivske pesmi ter otroške pesmi. Po vsebini ljudske pesmi ločimo tudi na t.i. variantne tipe. Variantni tip definira skupino pesmi, ki imajo podobno ali enako vsebino. Lahko ima poljubno število različic, ki jih imenujemo variante. Variante se med sabo lahko razlikujejo po narečju, deloma spremenjeni vsebini ali obliki.

Nekateri ločijo od ljudskih še ponarodele pesmi, kar pomeni, da se ve, kdo jo je napisal. Vendar pa je vsako pesem nekdo napisal, torej ni razloga, da ne bi ponarodelih pesmi šteli med ljudske. [10]

2.3.2 Struktura

Ritem v ljudskih pesmih nastane z izmenjavanjem poudarjenih in nepoudarjenih zlogov - zlogovni ritem. Vrste poudarka tudi ne imenujemo stopica, temveč verzni obrazec. Melodija vpliva na to, da se v ljudski pesmi ne pojavljajo tipične stopice umetne pesmi (jamb, amfibrah, anapest), temveč le trohej in daktil. Ločimo posamezne verze in verzne dvojice, distihe.

Če pesem uporablja posamezne verze, ti lahko sestavljajo nekitično pesem, medtem ko se verzne dvojice ponavadi združujejo v kitice. Poznamo vse od dvovrstičnih kitic do osemvrstičnih, najpogostejše pa so štirivrstične.

Različne ljudske pesmi imajo različne značilne dele, k so vezani na vsebino (npr. prošnja za dar pri kolednicah). Posebne oblike pesmi so še ponavljalna, kjer se del besedila velikokrat ponovi, ter naraščajoče in verižne, pri katerih se besedilo povezuje iz kitice v kitico. V slovenskih ljudskih pesmi je precej pogost tudi refren - ponavljajoč se del besedila, ki je lahko del verzov ali kitic, na začetku, v sredini ali na koncu. [10]

Poglavje 3

Opredelitev aplikacije

Aplikacija naj bo realizirana kot spletna stran. V njej je potrebno prikazati tri nivoje podatkov.

1. nivo - zajema vse variantne tipe, celotno bazo podatkov. Tipi naj bodo upodobljeni glede na podobnost med sabo, predvidoma na dvodimenzionalnem grafu.

2. nivo - zajema en variantni tip z vsemi pripadajočimi variantami. Prikazani naj bodo podatki o lokaciji in letnici variante.

3. nivo - besedilo ene variante.

Poglavje 4

Uporabljene metode in tehnologije

4.1 Programska oprema

Aplikacijo sem razvijala v operacijskih sistemih Windows 7 Professional in Ubuntu 14.04.3. Za programiranje sem uporabljala odprtokodni urejevalnik izvorne kode Geany. [12] Za poganjanje Python skript za obdelavo podatkov sem uporabljala ukazno vrstico Terminal. Aplikacijo sem testirala na LAMP strežniku nameščenem na Ubuntu sistemu.

Za pretvorbo nekaterih datotek s podatki (CSV, XML, JSON) sem uporabila nekaj javno dostopnih spletnih servisov.

4.2 Jeziki

Pri programiranju sem uporabila naslednje programske oziroma skriptne jezike.

4.2.1 HTML in CSS

HTML (HyperText Markup Language) [13] je označevalni jezik, ki se uporablja pri izdelavi spletnih strani. Z njim označimo osnovno vsebinsko strukturo strani, obenem pa tudi določimo semantični pomen elementov. CSS (Cascading Style Sheets) je opisni jezik, s katerim določimo izgled in obliko datoteke, zapisane v označevalnem jeziku. Lahko se aplicira na kakršenkoli XML dokument.

4.2.2 PHP

PHP (PHP: Hypertext Preprocesor) [14] je odprtokodni programski jezik, ki se ga uporablja za izdelavo dinamičnih spletnih strani. PHP deluje na strežniku, kjer dobi vhodno PHP kodo in iz nje generira spletno stran. PHP koda je lahko vključena v HTML, lahko pa je tudi v posebni datoteki. Sama sem PHP uporabila v manjši meri za razdelitev kode na manjše dele, ter za procesiranje podatkov.

4.2.3 Python

Python je splošen visokonivojski programski jezik [15], ki se uporablja za veliko različnih namenov. Je zelo kompakten in lahko berljiv za človeka. Ima dinamične podatkovne tipe in je priročen za delo s podatki. Uporabila sem ga za začetno obdelavo podatkov.

4.3 Javascript

Velik del aplikacije je napisan v JavaScriptu. JavaScript je objektni skriptni programski jezik, ki se ga uporablja pri izdelavi spletnih strani. [16] Uporablja se ga v kombinaciji s HTML-jem in se tako omogoči dinamičnost spletne strani.

Za JavaScript obstaja veliko knjižnic in ogrodij, izmed katerih sem uporabila spodaj naštete.

4.3.1 D3.js

D3.js (Data Driven Documents) je JavaScript knjižnica za vizualizacijo podatkov. [17] Z njo lahko naredimo dinamične in interaktivne vizualizacije. Uporablja SVG, HTML in CSS. SVG je ponavadi najpomembnejši element, saj tja vežemo podatke. Nad vizualizacijami lahko izvajamo prehode ("transitions"), kjer lahko spreminjamo vse lastnosti, ki so določene s HTML-jem, CSS-jem ali pa podatki samimi.

4.3.2 jQuery

jQuery je odprtokodna JavaScript knjižnica, ki olajša uporabo JavaScripta na strani klienta. [18] Je najbolj razširjena JavaScript knjižnica. Z njo se lažje izbira elemente DOM, izvaja AJAX klice in kreira odzive na različne akcije uporabnika.

4.3.3 Gmaps.js

Gmaps.js je JavaScript knjižnica, ki olajša uporabo Google Maps API-ja. [19] Google Maps APi dovoljuje spletnim razvijalcem, da v svoje strani vstavijo Googlove zemljevide. Gmaps.js omogoči bolj preprosto postavljanje markerjev na zemljevid in poenostavi delo s podatki.

4.3.4 AngularJS

AngularJS je odprtokodno ogrodje za spletne aplikacije, napisano v JavaScriptu. [20] Lahko ga uporabljamo skupaj s principom arhitekture uporabniških vmesnikov MVC. Najprej prebere HTML, v katerega so vključene posebne oznake, ki jih nato interpretira kot navodila za gradnjo strani.

4.3.5 Ostalo

Uporabila sem še majhen jQuery plugin jQuery.popupWindow in knjižnico numeric.js, ki jo je uporabila metoda MDS. [21]

4.4 MDS

Za delo z matriko podobnosti je bilo treba uporabiti enega izmed algoritmov, ki elemente, katerih faktor podobnosti je podan v matriki, postavi v 2D prostor glede na ta faktor. Uporabila sem algoritem MDS - multidimensional scaling, in sicer odprtokodno implementacijo v JavaScriptu (Ben Frederickson). [8]

Poglavje 5

Delo s podatki

5.1 Glavna zbirka podatkov

Večina podatkov, ki sem jih potrebovala, je bila vsebovana v XML datoteki. Razdeljena je na pet tematsko ločenih knjig ("book"), katere nato vsebujejo variantne tipe ("variant"), te pa vsebujejo variante ("doc"). Variantni tipi imajo naslov in zaporedno število, variante pa zaporedno število v okviru tipa, regijo, kraj, leto, meta podatke, originalno besedilo in lematizirano besedilo. Pri nekaterih variantah so podatki nepopolni: manjkajo leta, kraji pogosto niso specificirani podrobneje kot samo z imenom regije.

Spodaj je začetek XML datoteke, za boljšo predstavo o sestavi. (Šumniki so odstranjeni za potrebe LaTeX-a).

```
<books>
  <book>
    <no>1</no>
    <title>SLP 1: PRIPOVEDNE PESMI</title>
    <variant>
      <no>1</no>
      <title>1. PEGAM IN LAMBERGAR
```

```
    (DVOBOJ JUNAKA Z VELIKANOM)</title>
    <doc>
      <title>1.</title>
      <region>Gorenjsko</region>
      <place>Gorenjsko</place>
      <year>1775</year>
      <meta>
              Kraj: Gorenjsko (?)
              Z ps p. Dizma (Jozef) Zakotnik,
              ok. 1775 zapis se je z vso
              Zakotnikovo zbirko vred izgubil.
              Vsebino besedila moremo priblizno
              razbrati iz Linhartove nemske
              prepesnitve, ki jo podajamo v
              slovenskem izvlecku:
      </meta>
  <text>
              // lematizirano besedilo
  </text>
      <origtext>
              // originalno besedilo
      </origtext>
  <meta>
      </meta>
```

5.2 Matrika podobnosti

Matrika podobnosti je tekstovna datoteka. Vsebuje matriko, v kateri so zapisane vrednosti, ki predstavljajo podobnost med parom variantnih tipov. Matrika je bila izračunana iz baze podatkov, v kateri so bili za celoten korpus izračunani motivi. Iz teh podatkov se je izračunala distribucija variantnih

tipov prek teh motivov; določena je bila s povprečjem distribucij posameznih variant. Podobnost med variantnimi tipi je bila nato izračunana z metodo kosinusne podobnosti. [22] Podatki so na voljo o skoraj vseh variantnih tipih, ki jih vsebuje XML datoteka. Matriko sem pretvorila v format CSV, saj se je izkazalo, da je to v PHP-ju bolj preprosto.

5.3 Začetna obdelava podatkov s Pythonom

V začetku, še pred izdelavo načrta, sem s Pythonom napisala manjše število skript, s katerimi sem iz XML datoteke dobila števila določenih elementov. Za izdelavo prvega nivoja aplikacije je bil potrebno vedeti, koliko je najmanj in največ variant na en tip, zato da sem lahko določila, če se bo število variant na tem nivoju dalo prikazati oziroma na kakšen način.

V podatkih sem poiskala vse različne regije, ki se pojavljajo, ker je bil ta podatek pomemben za implementacijo drugega nivoja. V poštev so prišli tudi podatki o količini variant na leto v okviru posameznega variantnega tipa.

Izdelala sem tudi seznam variantnih tipov, kjer so bili shranjeni podatki o zaporedni številki, naslovu, številu variant in pa številki knjige, pod katero spadajo. Seznam sem shranila v format CSV, nato pa pretovrila v JSON, ki je bolj primeren za uporabo z D3.js.

Veliko XML datoteko sem s Python skripto razdelila na manjše datoteke, v katerih so bili shranjeni podatki o posameznem tipu. To je zmanjšalo čas nalaganja drugega in tretjega nivoja.

Spodaj je format datoteke JSON s seznamom variantnih tipov.

```
[
    {
        "st": "1",
        "naslov": "Pegam In Lambergar
                  (dvoboj junaka z velikanom)",
        "book": "1",
        "stv": "11"
    },
    {
        ...
    }
]
```

5.4 Obdelava v okviru aplikacije

Pred samim začetkom implementacije sem imela torej XML datoteko z vsemi podatki vseh variantnih tipov, matriko podobnosti in JSON datoteko s seznamom variantnih tipov. V aplikaciji je bilo nato potrebno ustvariti manjšo matriko podobnosti, ki vsebuje podatke samo za kroge, ki jih želimo prikazati naenkrat. (O tem več v poglavju 6.) Na prvem nivoju tako dostop do XML datoteke ni bil potreben, na drugem in tretjem pa se dostopa do majhnih XML datotek, ki vsebujejo informacije o enem variantnem tipu.

Poglavje 6

Postopek načrtovanja

Pred samo implementacijo izdelka je bilo potrebno izdelati načrt aplikacije. Ta načrt je vseboval funkcionalnosti in približno grafično podobo. Tekom implementacije je služil za oporo in vodilo, na nekaterih mestih pa sem se iz različnih razlogov tudi oddaljila od nameravanega načina izdelave. Načrt je pripomogel tudi k bolj jasni predstavi, kako naj bi ta aplikacija sploh izgledala.

6.1 Razdelitev na nivoje

Zaradi narave podatkov je bilo smiselno aplikacijo razdeliti na več nivojev. Na prvem nivoju bi bili tako prikazani variantni tipi, na drugem variante posameznega tipa in na tretjem besedilo posamezne variante. Preko višjih nivojev (če gledamo iz perspektive drevesne strukture) se torej dostopa do nižjih.

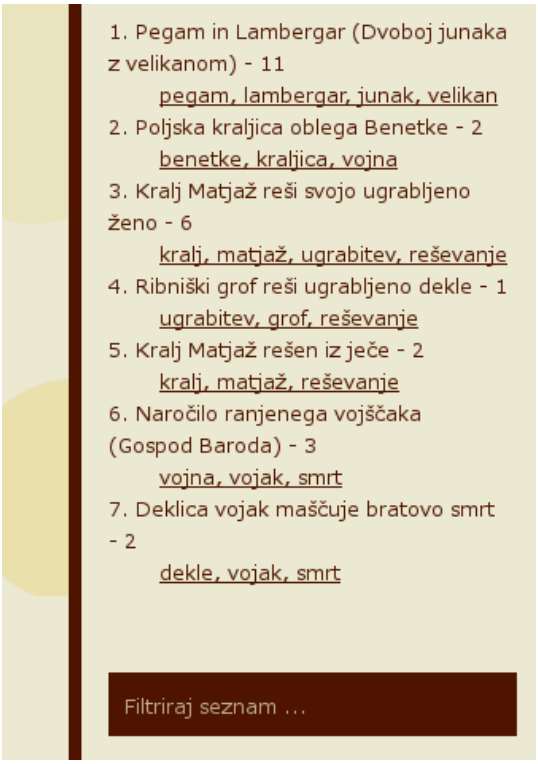
Načrt je bil definiran z opisom funkcionalnosti in s slikami posameznih korakov, kakor naj bi izgledali v delujoči aplikaciji.

Slika 6.1: Prvotna ideja o prikazu variantnih tipov

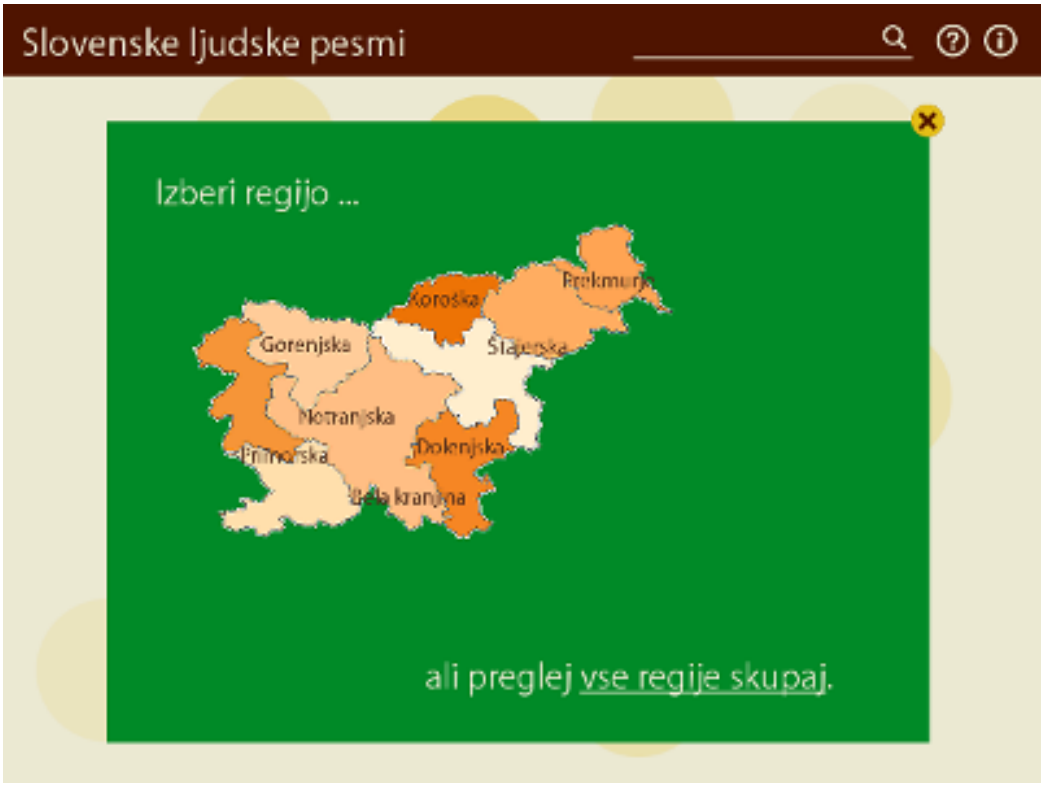
6.2 Prvi nivo - Variantni tipi

Nivo variantnih tipov je moral omogočiti izbiro posameznega variantnega tipa ter prikazati temu najbolj podobne tipe. Osnovni podatki o variantnem tipu so naslov, zaporedna številka, število variant in ključne besede. Odločila sem se, da bodo prikazani s krogi. Uporabnik bi imel nato možnost klika na posamezen krog, in graf naj bi se preuredil, tako da bi kazal v središču izbrani tip, okrog njega pa najbolj podobne tipe. (Prikazuje Slika 6.1)

Prvi nivo bi vključeval tudi osnovno iskanje po naslovih tipov. Seznam vseh tipov bi bil prikazan poleg grafa, in bi se sproti filtriral ob pisanju v ukazno vrstico. (Prikazuje Slika 6.2)



Slika 6.2: Seznam s filtriranjem

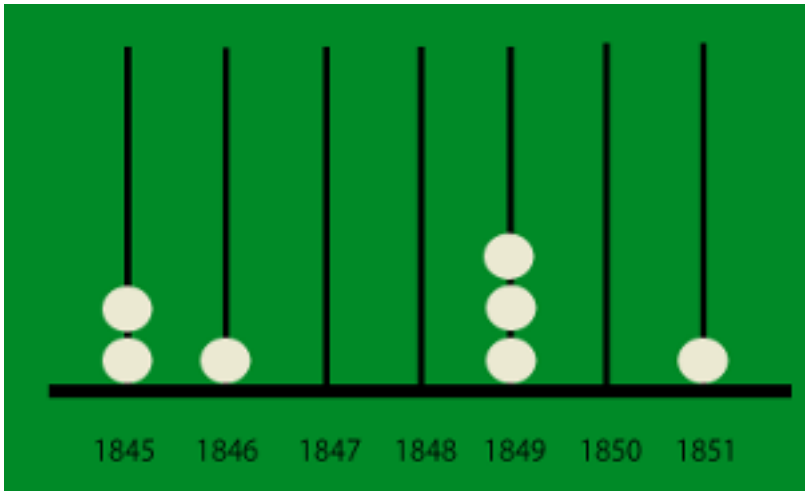


Slika 6.3: Prikaz regij, ki ni bil uporabljen

6.3 Drugi nivo - Variante

Drugi nivo bi se prikazal ob kliku na gumb ali povezavo v krogu, ki pripada določenemu tipu. Ker so bili na voljo podatki o kraju ter času, je bilo smiselno prikazati oba parametra. Najprej se mi je zdelo dobro, da bi uporabniku najprej omogočili izbiro regije (Slika 6.3), ter nato pokazali časovno razporejenost, vendar je bil končni plan tak, da bo možno videti oboje hkrati.

Postavitev določene variante v času bi bila prikazana z grafom, kjer imamo na y osi čas, na x pa število variant. Uporabila bi točkovni graf namesto stolpičnega grafa, da se lahko vidi individualno varianto. (Slika 6.4)



Slika 6.4: Prikaz variant določenega tipa po času

6.4 Tretji nivo - Besedilo

Ob kliku na posamezno varianto na točkovnem grafu ali zemljevidu se v pojavnem oknu prikaže besedilo te variante. Odločila sem se, da je za prikaz dovolj originalni tekst, saj je lematizirano besedilo bolj namenjeno iskanju. V začetku je bil načrt podrobneje razdelati besedilo, vendar se je izkazalo, da to ne bi bilo praktično, ker podatki niso pripravljeni na to in bi porabila preveč časa. (Več o tem v poglavju 7.)

6.5 Oblikovanje

Velik del izgleda in uporabnosti aplikacije predstavlja tudi oblikovanje. [9] Potrebno je poskrbeti, da so opcije za uporabnika dovolj očitne, da se elementi med sabo skladajo in da je izgled prijazen očem. Ker sama nimam veliko oblikovalskih izkušenj, sej izbrala precej preproste oblikovalske elemente in postavitve, za katere upam, da vseeno opravijo svojo funkcijo.

Barve sem izbrala glede na temo. Tople barve ter zelena, ki predstavlja

naravo, asociirajo na domače okolje in podeželje, ki si ga mnogo ljudi slika kot ozadje za petje ljudskih pesmi.

Poglavje 7

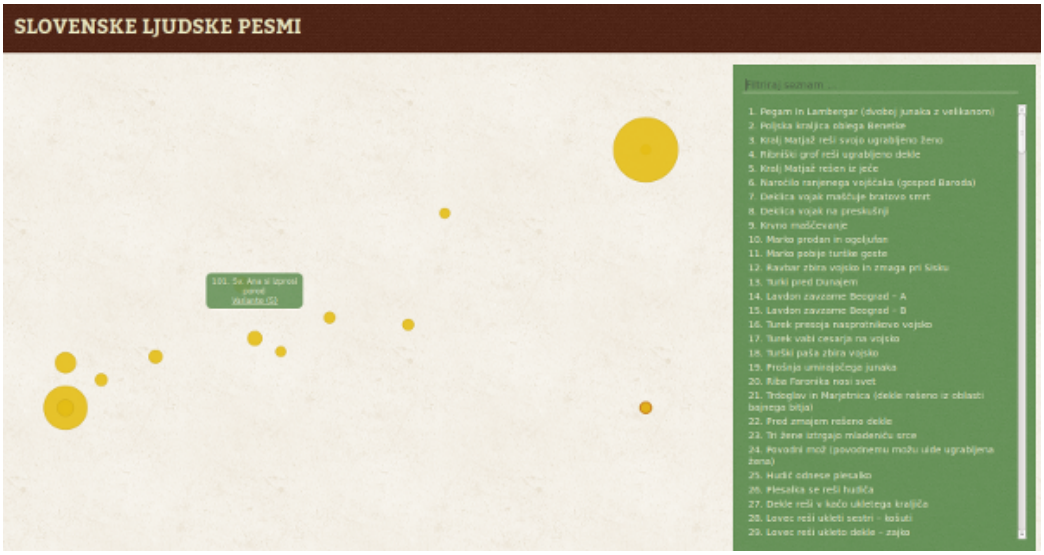
Implementacija

Implementacija je sledila strukturi, ki jo je zastavil načrt; najprej sem torej implementirala prvi nivo, in tako naprej. S HTML-jem in CSS-jem sem najprej zgradila osnovni izgled strani. Nisem se odločila za uporabo Bootstrap ogrodja, saj je bila struktura tako preprosta, da je bilo lažje in hitreje napisati svoj CSS.

7.1 Prvi nivo

V PHP-ju sem napisala funkcijo, ki dobi za parameter zaporedno številko variantnega tipa, na podlagi tega poišče v matriki podobnosti najbolj podobne tipe, ter naredi manjšo matriko podobnosti, ki vsebuje samo teh n tipov. Spremeljivka n je bila večinoma nastavljena nekje med 15 in 20. Poleg matrike vrne še polje tipov, ki so vključeni v matriko.

PHP skripta te podatke shrani v JSON datoteko, do katere dostopa JavaScript. V JavaScriptu sem napisala funkcijo, ki kliče PHP skripto, nato pa še drugo funkcijo, ki nariše graf (D3.js). Ta funkcija obstaja v dveh verzijah. Prva verzija se izvede ob nalaganju strani, druga pa vsakič, ko izberemo določen tip, ki trenutno ni izbran (če seveda vmes ne osvežimo strani). Prva funkcija tako vsebuje vse podatke, ki so potrebni za risanje vizualizacije,



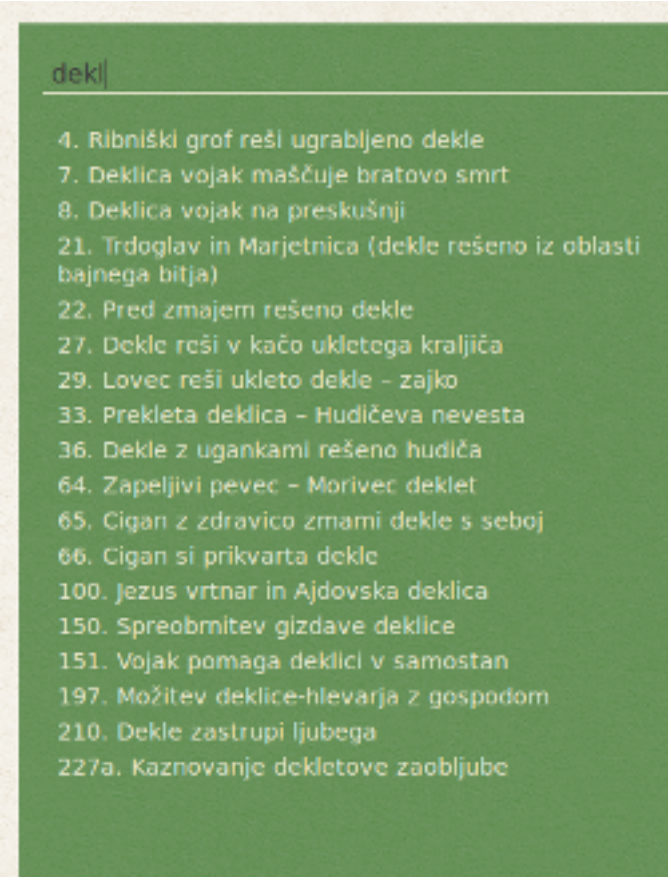
Slika 7.1: Zaslonska maska prvega nivoja

druga pa samo tiste, ki se spremenijo. (Graf prikazan na Sliki 7.1.) Krogi so veliki sorazmerno s številom variant, ki jih imajo.

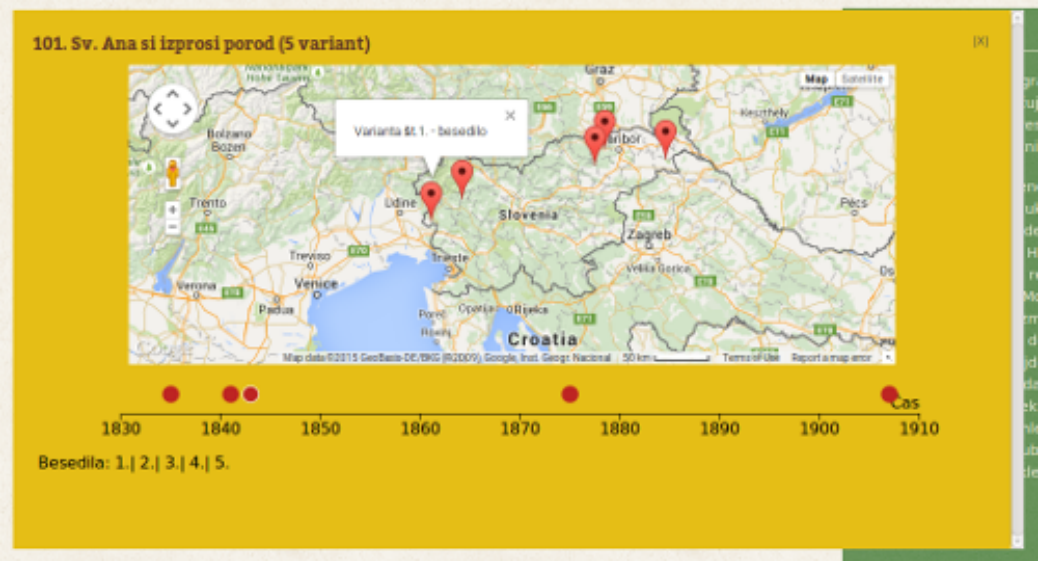
Podatki, ki se shranijo v SVG element kroga so zaporedna številka, koordinate, radij, število variant in naslov.

Preden se nariše graf, je potrebno iz matrike podobnosti izračunati koordinate. Za to uporabim implementacijo algoritma Multi-Dimensional Scaling, ki jo je napisal Ben Frederickson. [8] Nato je treba te koordinate pripraviti za risanje grafa. Funkcija mds namreč vrne zelo majhne koordinate, ki jih je potrebno normalizirati. Izvede se tudi t.i. tresenje koordinat, ki v naključnih smereh minimalno spremeni postavitev točk, zato da se prepreči prekrivanje.

Definiranih je tudi več jQuery funkcij, ki se odzivajo na akcije uporabnika (events); premik miške na krog bo pokazal tooltip z informacijami, klik pa poklical metodo za izračun nove matrike in nato na novo narisal graf.



Slika 7.2: Seznam implementiran z AngularJS



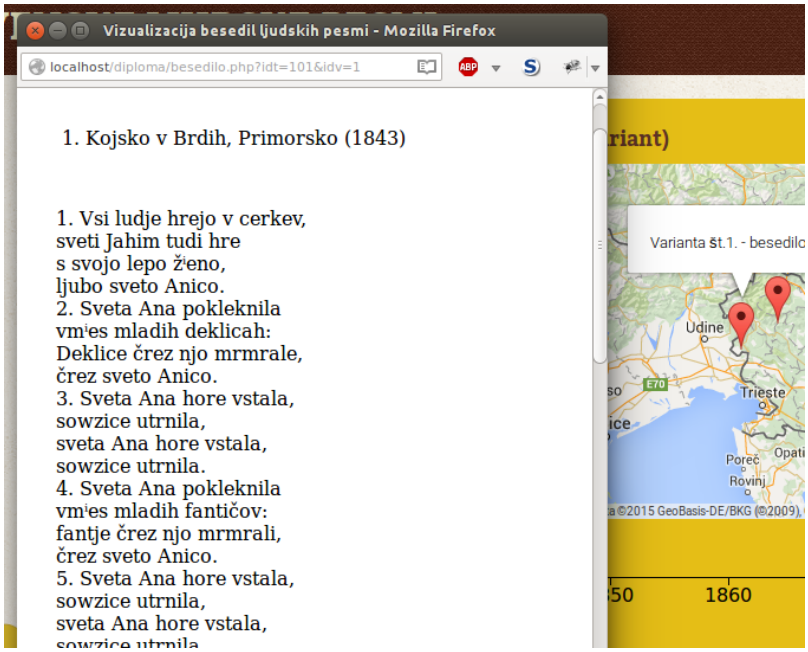
Slika 7.3: Pojavno okno z variantami

Seznam, ki je na prvem nivoju spletne aplikacije na desni (na Sliki 7.2), je realiziran z ogrodjem AngularJS, ki je bil uporabljen prav za ta namen. Seznam se filtrira s pisanjem v okence, s klikom na povezavo pa se zgodi enaka akcija kot ob kliku na krog. AngularJS bere podatke iz JSON datoteke, ki vsebuje seznam variantnih tipov.

7.2 Drugi nivo

Nivo variant se doseže s klikom na povezavo "Variante" v okencu, ki se pojavi nad krogom. Odpre se pojavno okno (Slika 7.3). Z jQueryem se dostopa do XML datoteke, kjer poiščemo izbrani tip. Na podlagi teh podatkov se izrišeta zemljevid z uporabo gmaps.js in graf (točkovni graf) z uporabo D3.js.

S klikom na posamezen marker na zemljevidu se obarva pripadajoča pika na točkovnem grafu, ki nam da informacijo o času, prav tako pa se ob kliku na piko na grafu označi pripadajoča varianta na zemljevidu (odpre se infor-



Slika 7.4: Besedilo variacije

mativno okno).

7.3 Tretji nivo

Zaradi smernic Google Maps se jQuery funkcije, ki jih želimo dodeliti elementom v tooltipu, zbrišejo, zato mi žal ni uspelo implementirati kode, ki bi odprla pojavno okno z besedilom. Zato so pod grafom povezave na različne variacije označene s številkami. Ob kliku na povezavo se odpre pojavno okno z besedilom variacije (Slika 7.4).

7.4 Oblikovanje

V oblikovanju se nisem veliko oddaljila od načrta. Nekateri elementi niso bili implementirani, manjše razlike so v barvah, uporabila pa sem tudi nekaj tekstur. [3]

Poglavje 8

Sklepne ugotovitve

8.1 Možne izboljšave

Narejena aplikacija ima osnovne funkcionalnosti za ogled baze besedil, vendar pa bi lahko uporabnost močno izboljšala z različnimi drugimi funkcijami. Najpomembnejša izmed teh je iskanje po vseh besedilih, ne samo naslovih variantnih tipov. Iskanje je bilo prvotno vključeno v načrt implementacije, vendar mi ga ni uspelo narediti. Iskanje bi moralo zajemati tako orignialna kot lematizirana besedila, saj uporabnik ne pozna vedno oblike besede, ki je v pesmi, ki jo išče, ali pa ga zanimajo besedila na določeno temo.

V začetku priprave na diplomsko delo sem dosti razmišljala tudi o podrobni razdelavi na nivoju posameznega besedila, vendar bi to zahtevalo precej več časa. Pred implementacijo takšne funkcionalnosti bi bila potrebna tudi temeljita obdelava podatkov, če že ne do nivoja posameznih črk, pa zagotovo zlogov. Omogočila bi prikaz kitic, rim, verznih obrazcev in informacij o njih. V bolj podrobni podatkovni zbirki bi lahko uporabili tudi podatke, ki določene oblike besedil vežejo na posamezna področja, in primerjavo med njimi. Tudi primerjava med dvema variantama istega variantnega tipa bi lahko bila zanimiva.

Zdi se mi, da bi bilo smiselno raziskati, kaj so informacije in povezave, ki zanimajo strokovnjake na tem področju, in potencialno dobiti nove mogoče smeri razvoja. Ker pa je smisel tega projekta tudi pritegniti zanimanje laične javnosti, bi bilo vredno razmisliti, kakšne funkcionalnosti bi bile privlačne v njihovem pogledu. V vsakem primeru bi bilo po mojem mnenju vredno združiti bazo besedil z bazo melodij in omogočiti sortiranje in prikaz tudi na glasbenem nivoju.

8.2 Zaključek

V tem delu je opisan celoten postopek izdelave aplikacije za vizualizacijo besedil ljudskih pesmi. Vključuje torej načrtovanje, obdelavo podatkov, uporabljena orodja in implementacijo. V diplomo sem vključila tudi nekaj o temah, ki ju delo združuje: ljudskih pesmih in vizualizaciji podatkov. Opisala sem zgodovino in lastnosti ljudskih pesmi, potem pa sem se osredotočila na zbiranje, ki je predstavljalo pomembno osnovo, ki mi je omogočila izdelavo diplome na to temo. Orisala sem tudi pomen vizualizacije podatkov na splošno.

Težave, ki sem jih imela pri izdelavi diplomskega dela, izvirajo iz nepoznavanja tehnologij in orodij in neizkušenosti pri delu na podobnih projektih in "user experience"oblikovanju. Aplikacija je brez iskanja po besedilih relativno pomanjkljiva, dalo pa bi se izboljšati tudi uporabnost, izgled in tek delovanja. Če bi se implementacije lotila še enkrat, bi se najbrž poskusila omejiti na en JavaScript framework, saj mi je nekompatibilnost povzročala nemalo preglavic.

Izdelava diplomskega dela je bila vsekakor učna izkušnja zame in upam, da bom lahko v prihodnosti delala na podobnih projektih. V tem poglavju sem opisala tudi možne izboljšave, ki, po mojem mnenju, skupaj z aplikacijo predstavljajo dobro osnovo za nadaljnji razvoj.

Literatura

[1] Rtv: Slovenska zemlja v pesmi in besedi. http://radioprvi.rtvslo.si/slovenska-zemlja-v-pesmi-in-besedi/, 2015. Dostopano: 7.9.2015.

[2] Sazu: Glasbenonarodopisni inštitut. http://gni.zrc-sazu.si/sl/predstavitev#v, 2015. Dostopano: 7.9.2015.

[3] Subtle patterns. http://subtlepatterns.com/, 2015. Dostopano: 7.9.2015.

[4] Wiki: Data visualization. https://en.wikipedia.org/wiki/Data_visualization, 2015. Dostopano: 8.9.2015.

[5] Wikipedia: Folk music. https://en.wikipedia.org/wiki/Folk_music, 2015. Dostopano: 6.9.2015.

[6] Mark Barrenechea. Big data: Big hype? http://www.forbes.com/sites/ciocentral/2013/02/04/big-data-big-hype/, 2013. Dostopano: 1.9.2015.

[7] Dorie Clark. Four things you need to know in the big data era. http://www.forbes.com/sites/dorieclark/2013/08/08/four-things-you-need-to-know-in-the-big-data-era/, 2013. Dostopano: 1.9.2015.

[8] Ben Frederickson. Mds implementation in javascript. http://www.benfrederickson.com/multidimensional-scaling/, 2015. Dostopano: 21.8.2015.

[9] Robert Klanten. *Data Flow: visualising information and graphic design.* Berlin : Gestalten, 2011, cop. 2008, 2011.

[10] Zmaga Kumer. *Slovenska ljudska pesem: njena vsebinska, oblikovna in glasovna podoba.* Ljubljana: Slovenska matica, 2002.

[11] Dušica Kunaver; Zmaga Kumer; Helena Ložar-Podlogar. *Pesmi in šege moje dežele.* Ljubljana: Državna založba Slovenije, 1987.

[12] Geany. `http://www.geany.org/`, 2015. Dostopano: 7.9.2015.

[13] Wiki: Html. `https://en.wikipedia.org/wiki/HTML`, 2015. Dostopano: 7.9.2015.

[14] Php.net. `http://php.net/`, 2015. Dostopano: 8.9.2015.

[15] Python.org. `https://www.python.org/`, 2015. Dostopano: 8.9.2015.

[16] W3schools: Js. `http://www.w3schools.com/js/`, 2015. Dostopano: 8.9.2015.

[17] D3.js. `http://d3js.org/`, 2015. Dostopano: 14.8.2015.

[18] jquery. `http://d3js.org/`, 2015. Dostopano: 4.6.2015.

[19] gmaps.js. `https://hpneo.github.io/gmaps/`, 2015. Dostopano: 30.8.2015.

[20] Angularjs. `https://angularjs.org/`, 2015. Dostopano: 23.8.2015.

[21] jquery.popupwindow. `http://swip.codylindley.com/popupWindowDemo.html`, 2008. Dostopano: 30.8.2015.

[22] Matija Strle, Gregor in Marolt. Uncovering semantic structures within folk song lyrics, 2014.

[23] Marko Terseglav. *Ljudsko pesništvo.* Ljubljana: Državna založba Slovenije, 1987.

[24] Jože Glonar; Karel Štrekelj. *Slovenske narodne pesmi*. Ljubljana: Cankarjeva založba, 1980.